



# pyJAliEn client

Adrian Sevcenco, ISS, RO

[https://github.com/adriansev/jalien\\_py](https://github.com/adriansev/jalien_py)



# Outline



- **Why**
- **How**
  - Python
  - Websockets
  - Asyncio
  - JSON messages
- **What (can it do)?**
  - Metadata related manipulation
  - Job Management
  - File operations

# Why

- New Java based central services – JAliEn
  - There is an jalien.sh client (Java based)
  - Alternative client needed for flexibility - alien.py
    - 26k python script (7.6k in minified compressed form)
    - but lacking interactivity features (terminal interaction)
- The new JAliEn services offer websocket endpoints for communication
  - Websocket is HTTP with some important extras:
    - Full duplex, connection-oriented communication
    - Stateful protocol
    - Optimized, low-latency communication

# How

- Python (3.6 or higher) (proper asyncio support and [other improvements](#))
  - Universal
  - Simple, natural language
  - Requirements : pyOpenSSL, websockets, pyxrootd
    - Waiting for upstream fixes to trickle to pypi repository
- Websocket connection
  - SSL connection to upstream
  - Serialized queries (send/receive dialogue)
  - Simple communication, optionally compressible

# How (cont.)

- Asyncio
  - Communication infrastructure (transport and protocol abstractions, multiplexing I/O access over sockets)
  - Single threaded concurrent code (synchronization primitives)
  - Coroutine\* paradigm (async def – coroutine functions, await – suspend execution coroutine completion until return data is fetch and returned)

\*Coroutine: functions/components that can be suspended/resumed – it can (a)wait for the results



# How (cont.)



- JSON
  - Easy serialization of “items” and concepts (dictionaries, lists, maps)
  - Native serialization/deserialization of python objects
  - Space efficient (w.r.t. xml)
  - Easier to use in developing

# What (can it do)?

- API/metadata operations
  - Catalog manipulation (files metadata) (examples in command mode)
    - `alien.py ls -l`
    - `alien.py mkdir my_new_dir`
    - `alien.py stat /my_alien_file`
  - Job management
    - `alien.py ps -A`
    - `alien.py kill <job_id>`

# What (can it do)?

- XRootD operations (cp)

- Copy from local to grid (upload)

```
alien.py cp file://testfile ./  
jobID: 1/2 ... STATUS: OK ; SPEED = 8.77 kB/s ; MESSAGE: [SUCCESS]  
jobID: 2/2 ... STATUS: OK ; SPEED = 21.27 kB/s ; MESSAGE: [SUCCESS]
```

Or

```
alien.py cp file://testfile %ALIEN/test_home  
jobID: 1/2 ... STATUS: OK ; SPEED = 4.82 kB/s ; MESSAGE: [SUCCESS]  
jobID: 2/2 ... STATUS: OK ; SPEED = 33.71 kB/s ; MESSAGE: [SUCCESS]
```

- “./” is converted to cwd both local and grid
- %ALIEN is converted to AliEn home directory (~ and # are interpreted by bash)
- Copy from grid to local (download)

```
alien.py cp test_home file://test_home2  
jobID: 1/1 ... STATUS: OK ; SPEED = 12.62 kB/s ; MESSAGE: [SUCCESS]
```

- A meta-url is created with all replicas
  - With default sources = 1 each source is tried until successful
  - Starting with XRootD 4.10 source > 1 will download in parallel
- both source and destination support the option of {src,dst},disk:N,SE1,SE2,!SE3
- NO grid to grid copy! (under discussion)



# What (can it do)?

- Shell interaction
  - 'Sh:' - will execute everything after the tag in the shell
  - '|' will pipe the result from the left side to the shell commands of the right side
- Command line mode and interactive mode
  - Anything after alien.py will be taken as JAliEn command
  - Interactive mode if there is no argument
- Low latency commands : one can chain multiple alien.py commands in a bash script

# Debugging

For those interested in under-the-hood mechanics:

- Environment variables (set to anything):
  - JALIENPY\_DEBUG : printouts of sent messages
  - JALIENPY\_XRDDEBUG : printouts of command and parameters related to XRootD commands
- Enabling debug by the script name
  - If script have `_json` in name : it will print raw json messages (full answer of the server)
  - If script have `_json_all` in name : it will print raw json messages (full answer of the server) including the metadata section of the answer

# Future development

- Recursive copy
  - Uploading/Downloading directories
- Smart interactive interface : history file, Up/Down, Left/Right keyboard input
  - Ådne Nerheim : will work on full interactive shell implementation in Python

# Feedback

- A feedback on what features/commands are missing at this point would be really useful
- [https://github.com/adriansev/jalien\\_py/issues](https://github.com/adriansev/jalien_py/issues)
- This is the best time for improvements/additions in both JAliEn server code and also in clients code

[Costin.Grigoras@cern.ch](mailto:Costin.Grigoras@cern.ch) - for JAliEn features

[Adrian.Sevcenco@cern.ch](mailto:Adrian.Sevcenco@cern.ch) \*- alien.py features

[\*] I prefer the [issues](#) page of the project – better tracking of things to solve

# Examples

```
alien.py stat /alice/cern.ch/user/a/asevcenc/testfile4
```

```
alien.py stat /alice/cern.ch/user/a/asevcenc/testfile4
File: /alice/cern.ch/user/a/asevcenc/testfile4
Type: f
Owner: asevcenc:asevcenc
Permissions: 755
Last change: 2019-04-01 12:10:01.0 (1554113401000)
Size: 0 (0 B)
MD5:
GUID: 4aa70d10-5466-11e9-8e10-d8d385976ec8
    GUID created on Mon Apr 01 12:09:52 CEST 2019 (1554113392737) by d8:d3:85:97:6e:c8
```

```
alien.py ls -l /alice/cern.ch/user/a/asevcenc/testfile4
```

```
alien.py ls -l /alice/cern.ch/user/a/asevcenc/testfile4
-rwxr-xr-x  asevcenc asevcenc          0 Apr 01 12:10  testfile4
```

```
alien.py find /alice/cern.ch/user/a/asevcenc test*
```

```
alien.py find /alice/cern.ch/user/a/asevcenc test*
/alice/cern.ch/user/a/asevcenc/testfile
/alice/cern.ch/user/a/asevcenc/testfile2
/alice/cern.ch/user/a/asevcenc/testfile3
/alice/cern.ch/user/a/asevcenc/testfile4
```

# Examples

Interactive mode examples

List first 3 jobs of user aliproduct

```
jsh: /alice/cern.ch/user/a/asevcenc/ > ps -u aliproduct | head -n 3
aliproduct 834810241  _____ SP      AOD_merge.sh
aliproduct 921515484  _____ SP      AOD_merge.sh
aliproduct 921538600  _____ D       AOD_merge.sh
```

Print/use an environment variable within shell

```
alien.py
jsh: /alice/cern.ch/user/a/asevcenc/ > sh:echo "$TMPDIR"
/home/adrian/tmp
```

Within client, list the files and then use shell constructs to filter and count

```
jsh: /alice/cern.ch/user/a/asevcenc/ > ls | grep '^test' | wc -l
2
```

Command mode, chain commands example:

For each jobID of user adobrin (but with a limit of 1) print the jdl

```
for id in $(alien.py ps -u adobrin -A -l 1 | awk '{print $2}'); do alien.py ps -jdl ${id}; done
InputDataListFormat = "xml-single";
Validationcommand = "/alice/cern.ch/user/a/adobrin/treePass118r/datapass1_validation.sh";
Executable = "/alice/cern.ch/user/a/adobrin/treePass118r/datapass1.sh";
```



# Feedback needed



[https://github.com/adriansev/jalien\\_py](https://github.com/adriansev/jalien_py)

And the pivotal point of any project

[https://github.com/adriansev/jalien\\_py/issues](https://github.com/adriansev/jalien_py/issues)

**THANK YOU !**